

PROYECTO FINAL REDES E INFRAESTRUCTURA



ANA CRISTINA QUINTERO
MARÍA FERNANDA TELLO
SEBASTIAN BELALCAZAR

ANÁLISIS

La base de datos de los estudiantes de East High. (Notas, premios obtenidos por estos estudiantes.) Es un dataset con notas de alrededor de 400 estudiantes, donde se encuentran los puntajes en pruebas de matemáticas, inglés y ciencias.

Decidimos utilizar la plataforma de procesamiento de datos distribuidos Apache Spark en este proyecto debido a su rendimiento superior, escalabilidad, flexibilidad, facilidad de uso

La aplicación debe poder generar reportes y visualizarlos en un dashboard.

La aplicación basada en microservicios y API Rest en un cluster de contenedores

Diagrama de los componentes a utilizar(Pipeline)

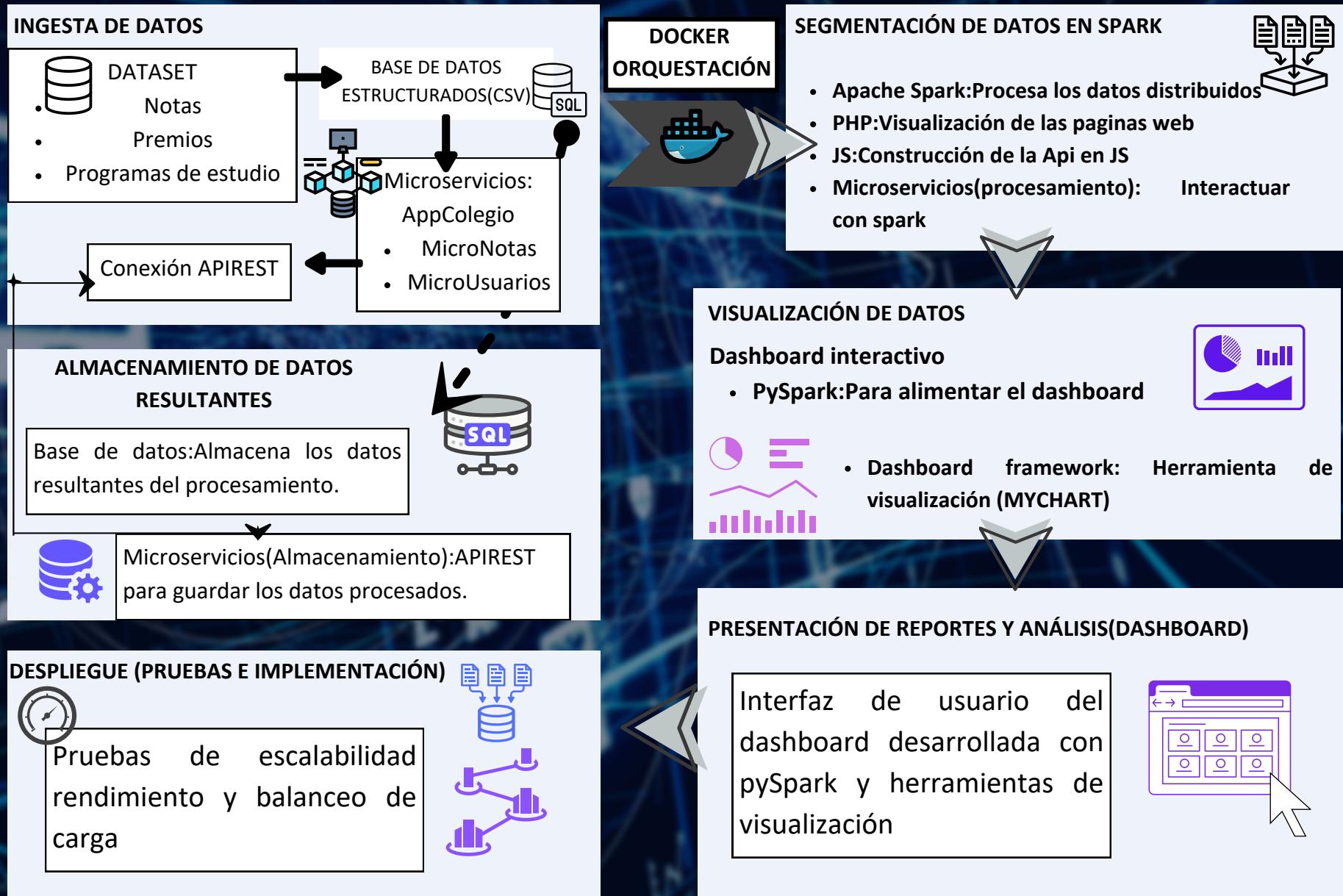
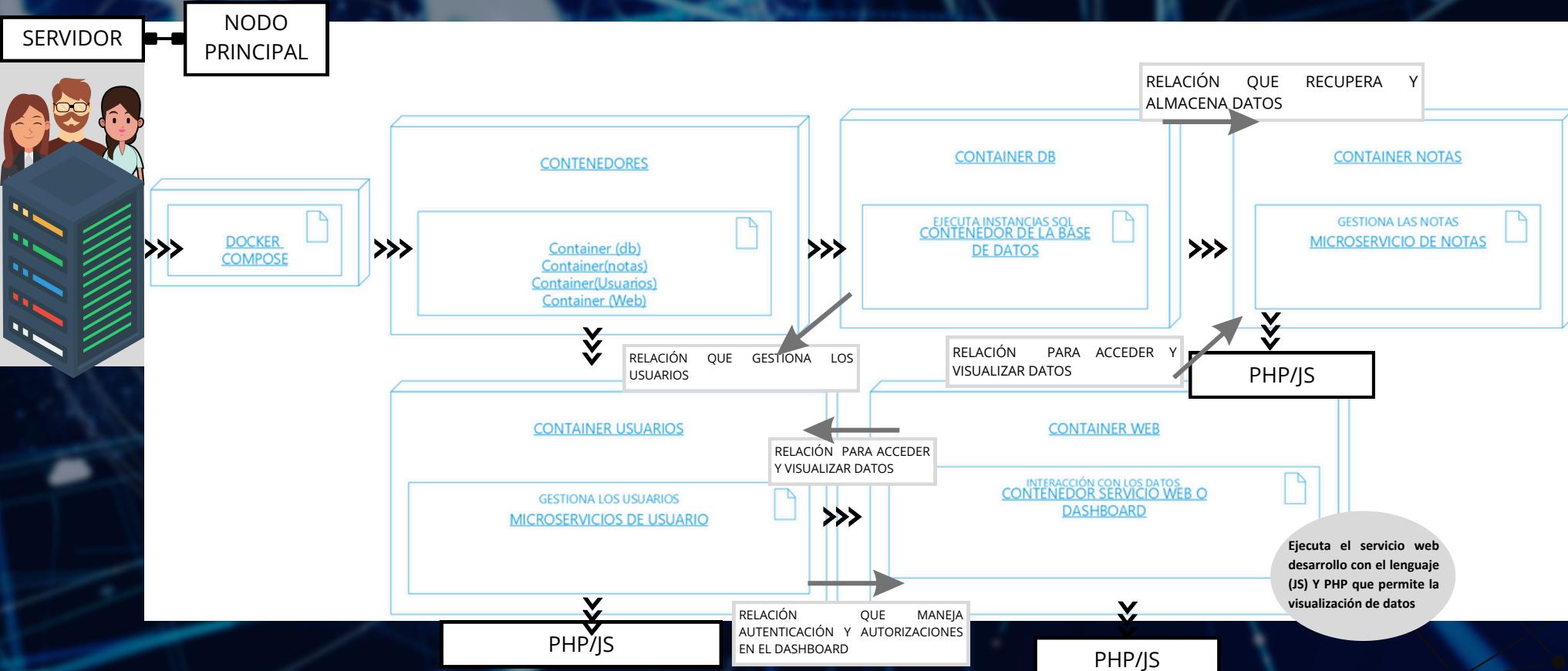


DIAGRAMA DE DESPLIEGUE



Implementación y Pruebas

Implemente la solución diseñada: se debe implementar la solución diseñada utilizando las tecnologías y herramientas seleccionadas

PRUEBAS CON PySpark

URL: spark://192.168.100.3:7077
Alive Workers: 0
Cores in use: 0 Total, 0 Used
Memory in use: 0.0 B Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 3 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

- Workers (0)

Worker Id	Address	State	Cores	Memory	Resources
-----------	---------	-------	-------	--------	-----------

▼ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

- Completed Applications (3)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20240524030628-0002	ConsoleAnalysis	0	2.0 GiB		2024/05/24 03:06:28	root	FINISHED	24 s

Using Python version 3.10.12 (main, Nov 20 2023 15:14:05)

Spark context Web UI available at <http://10.0.2.15:4040>

Spark context was created as 'sc' (master = local[*], app id = local-1716517781965).

Spark context available as 'sc'.
SparkSession available as 'spark'.

```
>>> df = spark.read.options(header='True', inferSchema='True').csv('/root/Caoskol-Project/US_Dataset.csv')
```

PYSPARK

appPySpark.py

```
root@servidorUbuntu: ~/Caoskol-Project

import sys
from pyspark.sql import SparkSession

def main():
    # Inicia la sesión de Spark
    spark = SparkSession.builder.appName("ConsoleAnalysis").getOrCreate()

    # Carga los datos
    df = spark.read.csv("/root/Caoskol-Project/US_Dataset_users.csv", header=True, inferSchema=True)

    # Realiza operaciones, por ejemplo, un agregado
    result_df = df.groupBy("grado").count()

    # Guarda los resultados en un archivo
    result_df.write.csv("/root/Caoskol-Project/results", mode="overwrite", header=True)

    # Detiene la sesión de Spark
    spark.stop()

if __name__ == "__main__":
    main()
```

PYSPARK

analysis.py

```
root@servidorUbuntu: ~/Caoskol-Project

import sys
from pyspark.sql import SparkSession
from pyspark.sql.functions import mean, stddev, min, max, avg, corr, col

def main():
    # Inicia la sesión de Spark
    spark = SparkSession.builder.appName("Data_Analysis").getOrCreate()

    # Carga los datos
    notas_df = spark.read.csv("/root/Caoskol-Project/US_Dataset.csv", header=True, inferSchema=True)

    # Imprimir los nombres de las columnas originales
    print("Nombres de columnas originales:", notas_df.columns)

    # Limpiar espacios en los nombres de las columnas
    notas_df = notas_df.toDF(*[c.replace(' ', '').replace("'", '') for c in notas_df.columns])

    # Verificar los nombres de las columnas después de la limpieza
    print("Columnas después de la limpieza:", notas_df.columns)

    # Limpiar espacios en los nombres de las columnas, incluyendo espacios internos y paréntesis
    notas_df = notas_df.toDF(*[c.replace(' ', '').replace("'", '').replace('(', '').replace(')', '') for c in notas_df.columns])

    # Verificar los nombres de las columnas después de la limpieza
    print("Columnas después de la limpieza:", notas_df.columns)

    # Calcular estadísticas descriptivas
    "análisis.py" 821 2406B
```

Promedio Matemáticas	Promedio Inglés	Promedio Ciencias	Promedio Premios	Desvío Estándar Matemáticas	Desvío Estándar Inglés	Desvío Estándar Ciencias	Desvío Estándar Premios	Máximo Premios	Mínimo Premios
60.3775	60.735	NULL	1.5725	23.92752200968185	22.087766945111348	NULL	NULL	1.	0829013983722113

24/05/24 12:45:00 INFO FileSourceStrategy: Pushed filters:

max_math	min_math	avg_math	max_awards	avg_awards
100.0	20.0	60.3775	3.0	1.5725

24/05/24 12:45:11 INFO CodeGenerator: Code generated in 4.546884 ms

grado	count	grado	Promedio Matemáticas	Promedio Inglés	Promedio Ciencias	
8.0	52	8.0	63.09615384615385	59.44230769230769	NULL	
7.0	76	7.0	57.85526315789474	60.671052631578945	NULL	
11.0	47	11.0	57.97872340425532	65.57446808510639	NULL	
10.0	61	10.0	58.78688524590164	61.114754098360656	NULL	
6.0	86	6.0	62.93023255813954	60.05813953488372	NULL	
9.0	48	9.0	62.166666666666664	58.291666666666664	NULL	
12.0	30	12.0	58.866666666666667	60.63333333333333	NULL	

Correlación Premios-Matemáticas	Correlación Premios-Inglés	Correlación Premios-Ciencias
-0.02141959461403...	-0.04184099492909237	NULL

Implementación y Pruebas

Implemente la solución diseñada: se debe implementar la solución diseñada utilizando las tecnologías y herramientas seleccionadas

Docker Swarm:

VM Principal (Cliente)

VM Secundaria (Servidor)

```
Node left the swarm.  
root@servidorUbuntu:~/Caoskol-Project# docker swarm init --advertise-addr 192.168.100.2  
Swarm initialized: current node (pmns33epz9p7ubtt6c37jmqkr) is now a manager.  
  
To add a worker to this swarm, run the following command:  
  
    docker swarm join --token SWMTKN-1-6b4e1h0tua65x50izdcybrbljn2dncii1nk0p6hctrqkr5aae1-5e1kho2hrxtkulj4u9gkxznx 192.168.100.2:2377  
  
To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.  
  
root@servidorUbuntu:~/Caoskol-Project# docker stack deploy -c docker-swarm.yml caoskol  
Since --detach=false was not specified, tasks will be created in the background.  


| Image               | Tag    | Status   | Created        | Size      |
|---------------------|--------|----------|----------------|-----------|
| seasbelmos/haproxy  | latest | Inactive | 9 minutes ago  | 38.51 MB  |
| seasbelmos/web      | latest | Inactive | 21 minutes ago | 177.19 MB |
| seasbelmos/notas    | latest | Inactive | 21 minutes ago | 406.3 MB  |
| seasbelmos/usuarios | latest | Inactive | 21 minutes ago | 406.3 MB  |


```

Implementación y Pruebas

Implemente la solución diseñada: se debe implementar la solución diseñada utilizando las tecnologías y herramientas seleccionadas

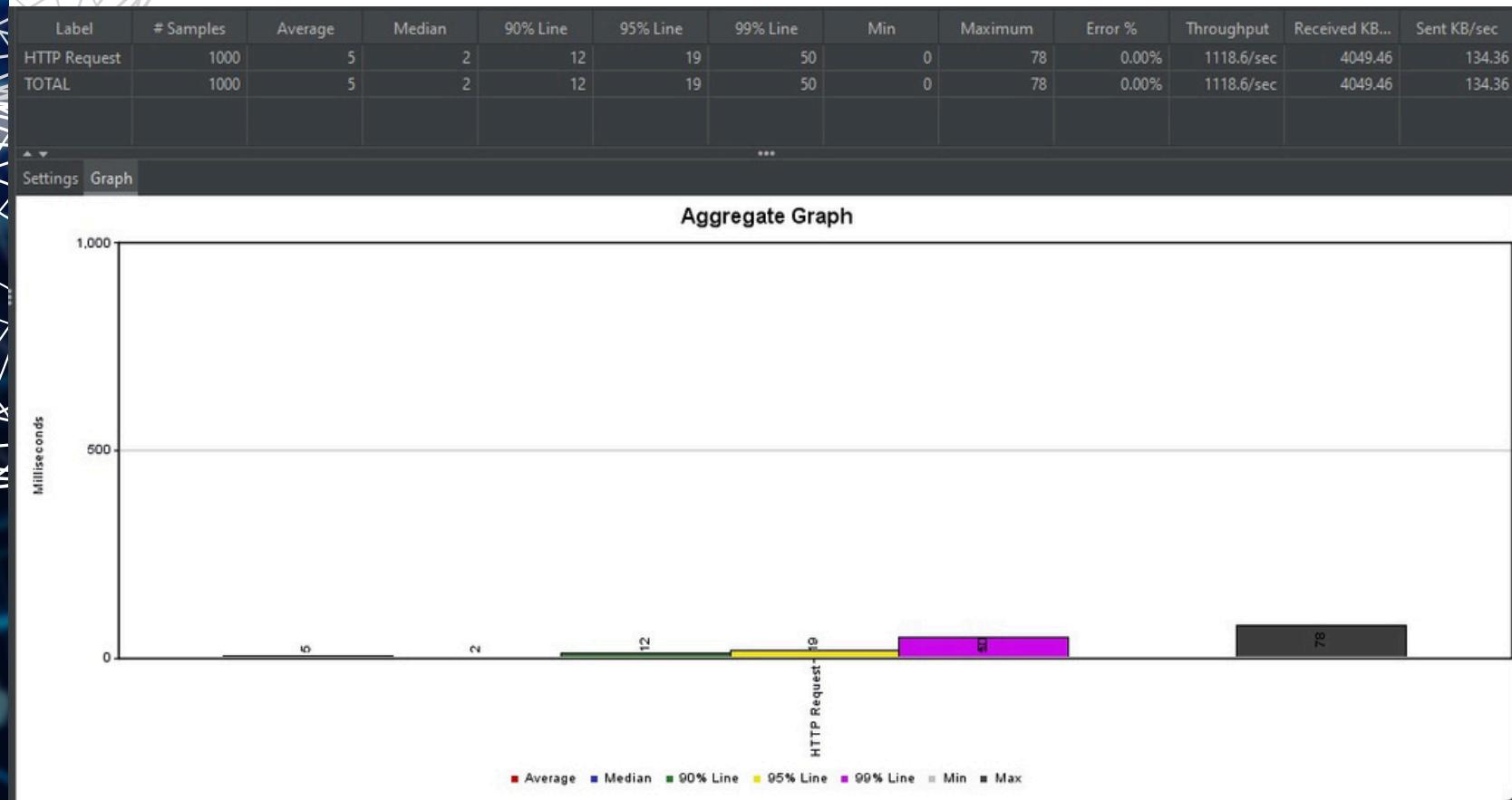
PRUEBAS CON JMETER

The screenshot shows the JMeter interface with a summary report configuration. The 'Basic' tab is selected. The 'Web Server' section has 'Protocol [http]:' set to an empty field, 'Server Name or IP:' set to '192.168.100.3', and 'Port Number:' set to '9080'. The 'HTTP Request' section has 'Method' set to 'GET' and 'Path:' set to an empty field. The 'Summary Report' section has 'Name:' set to 'Summary Report' and 'Comments:' set to an empty field. Below the configuration is a table showing test results:

Label ↓	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	10	6	1	48	13.81	0.00%	137.0/sec	495.92	16.45	3707.1
TOTAL	10	6	1	48	13.81	0.00%	137.0/sec	495.92	16.45	3707.1

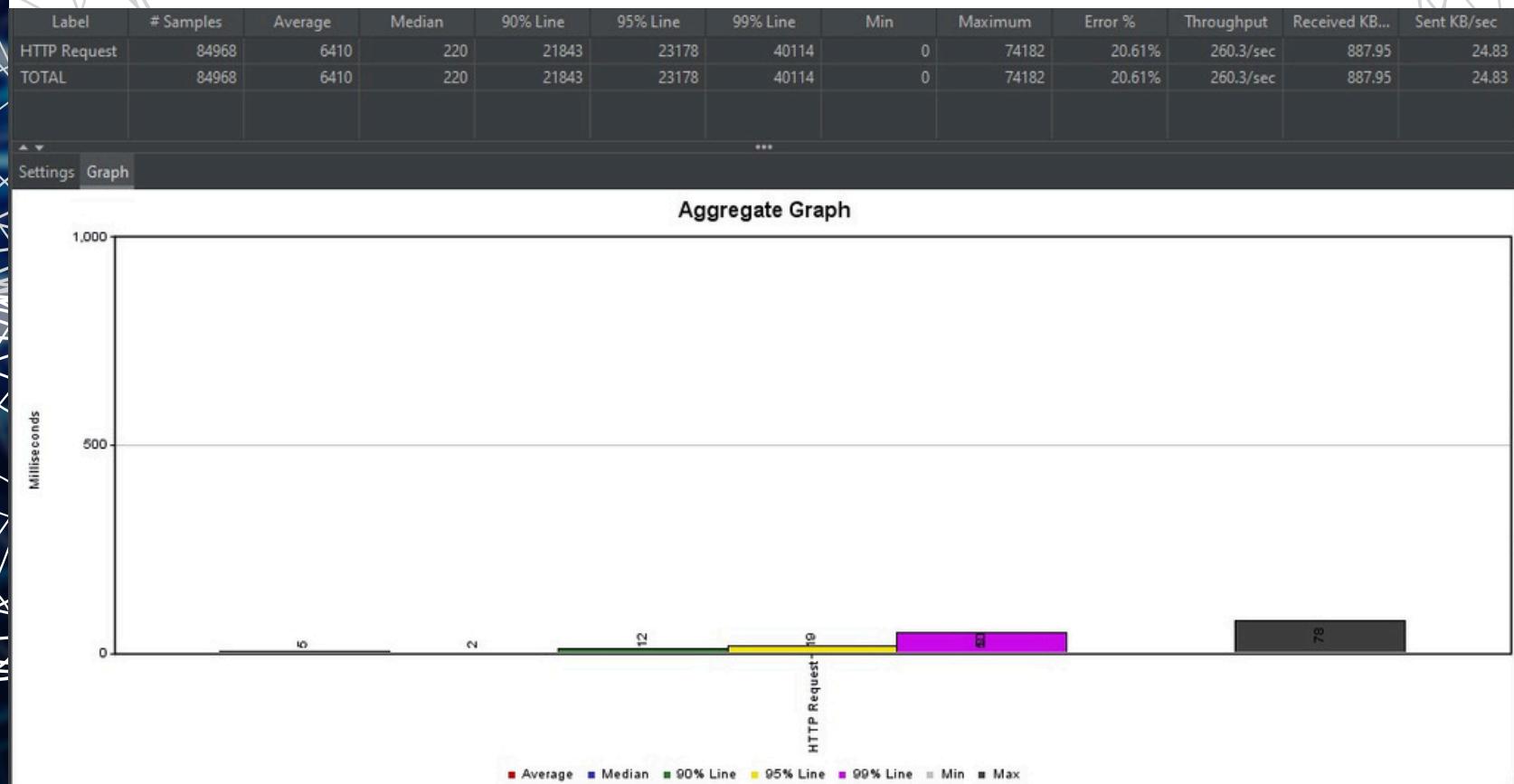
Implementación y Pruebas

PRUEBAS CON JMETER(100x10)



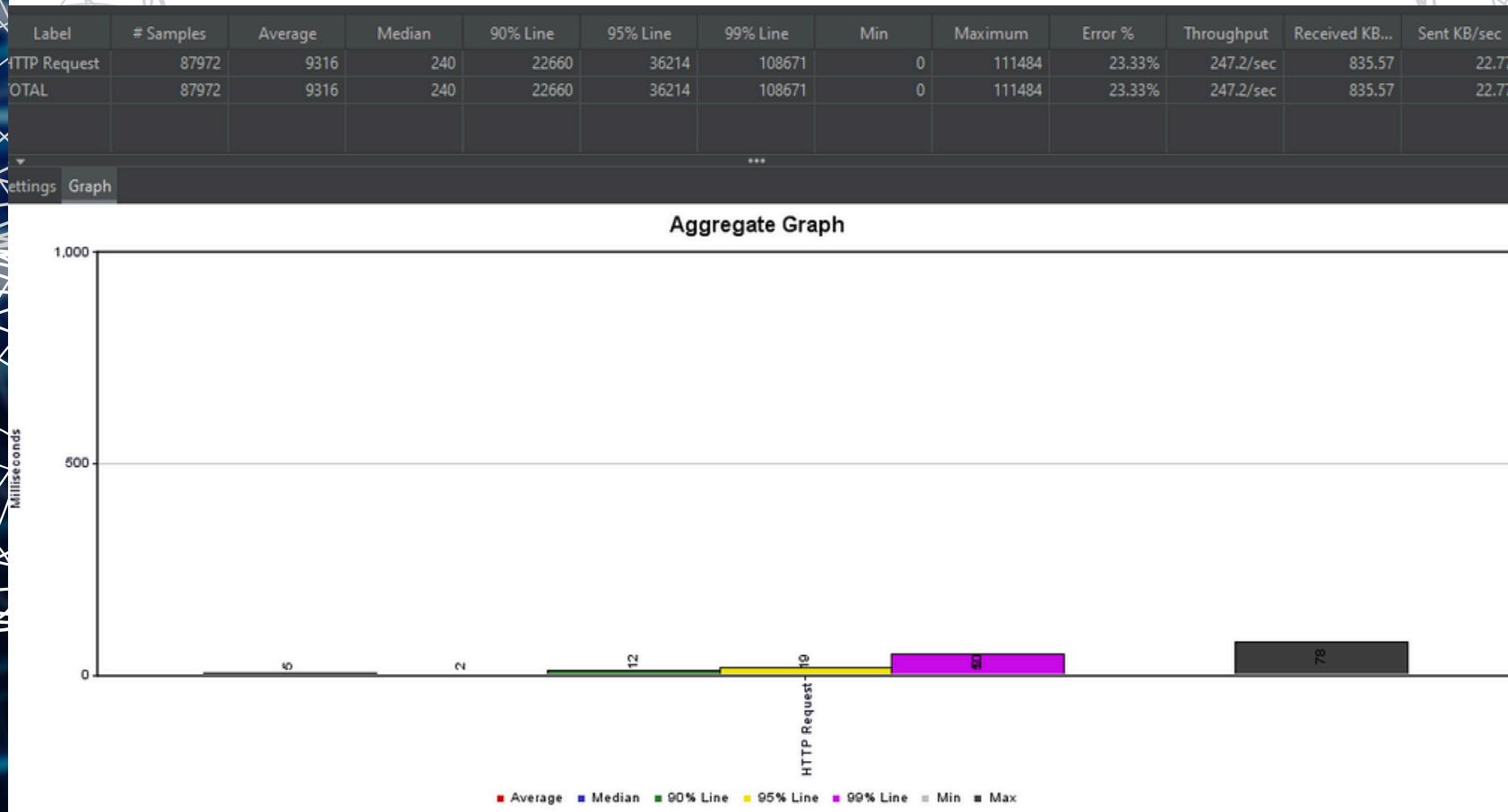
Implementación y Pruebas

PRUEBAS CON JMETER(10000x10)



Implementación y Pruebas

10000 x 10 a 10 escalas



Implementación y Pruebas

Implemente la solución diseñada: se debe implementar la solución diseñada utilizando las tecnologías y herramientas seleccionadas

Escalabilidad

```
root@clienteUbuntu:~/Caoskol-Project# docker service scale caoskol_web=10
caoskol_web scaled to 10
overall progress: 10 out of 10 tasks
1/10: running [=====>]
2/10: running [=====>]
3/10: running [=====>]
4/10: running [=====>]
5/10: running [=====>]
6/10: running [=====>]
7/10: running [=====>]
8/10: running [=====>]
9/10: running [=====>]
10/10: running [=====>]
verify: Service caoskol_web converged
```

Funcionalidad

A screenshot of a web browser window titled "Not secure | http://192.168.100.3:9080/index.html". The browser's toolbar includes icons for Keep, Speedtest, Test, Courses, Images, Books, GitHub, WPS, Wallet, Languages, LinkedIn, Movies, Tools, Math, Programming, and Social Media. The main content area displays the title "East High" and the subtitle "Resultados de Pruebas Anuales". A modal dialog box is centered on the page, titled "Iniciar Sesión". It contains two input fields: "ID" with the value "4" and "Clave" (password) with an empty field. A blue "Ingresar" (Enter) button is at the bottom of the modal.

CONCLUSIONES

Este proyecto demuestra la viabilidad de utilizar una infraestructura basada en microservicios y contenedores para el análisis de datos educativos. La utilización de Docker, Apache Spark y PySpark permite un procesamiento eficiente y escalable de grandes volúmenes de datos, facilitando la generación de reportes y su visualización en un dashboard interactivo.

APRENDIZAJES Y RESULTADOS CLAVE

- Escalabilidad y Eficiencia
- Procesamiento Distribuido
- Flexibilidad y Agilidad
- Automatización: